CLAIMS

1. A method of generating code for scheduling the execution of binary code translated from a source format to a target format, said method comprising the steps of:

    a) identifying a set of target instructions semantically equivalent to a given source instruction;

    b) analysing the set of target instructions to identify data dependencies in said target instructions;

    c) assigning an identifier to one or more of said target instructions for use by a code analyser in scheduling the processing of said set of target instructions in accordance with the identified data dependencies.

2. A method according to claim 1 in which the set of target instructions is identified in a translation template associated with a given source instruction said template being a component of a translator program for translating instructions in the source format into instructions in the target format.

3. A method according to claim 2 in which the analysis of the target instructions is carried out prior to the compilation of the translation templates into said translator program.

4. A method according to claim 2 in which the identifiers are assigned to said target instructions prior to the compilation of said translator program.

5. A method according to claim 1 in which said code analyser uses the identifiers for optimising the translated code for processing in a parallel processing environment.

6. A method according to claim 1 in which data dependencies are represented by a directed acyclic graph and the identifier is arranged to identify said dependency signalling an appropriate edge in the set of target instructions to said code analyser.

7. A method according to claim 2 in which each translation template is associated with a corresponding analysis routine for generating said code for scheduling the execution of said translated code.

8. Apparatus for generating code for scheduling the execution of binary code translated from a source format to a target format, said apparatus comprising:

    a) a set of target instructions semantically equivalent to a given source instruction;

    b) an instruction analyser for analysing the set of target instructions to identify data dependencies in said target instructions;

    c) a dependency identifier for assigning an identifier to one or more of said target instructions for use by a code analyser in scheduling the processing of said set of target instructions in accordance with the identified data dependencies.

9. Apparatus according to claim 8 in which the set of target instructions is identified in a translation template associated with a given source instruction said template being a component of a translator program for translating instructions in the source format into instructions in the target format.

10. Apparatus according to claim 9 in which the analysis of the target instructions is carried out prior to the compilation of the translation templates into said translator program.

11. Apparatus according to claim 9 in which the identifiers are assigned to said target instructions prior to the compilation of said translator program.

12. Apparatus according to claim 8 in which said code analyser uses the identifiers for optimising the translated code for processing in a parallel processing environment.

13. Apparatus according to claim 8 in which data dependencies are represented by a directed acyclic graph and the identifier is arranged to identify said dependency signalling an appropriate edge in the set of target instructions to said code analyser.

14. Apparatus according to claim 9 in which each translation template is associated with a corresponding analysis routine for generating said code for scheduling the execution of said translated code.

15. A computer program arranged to perform the method of claim 1.

16. A binary code translator for translating binary code from a source format to a target
format for execution on a target processor, the translator comprising:

    a) a set of translation templates, each template providing a set of target format
       instructions which together are semantically equivalent to an associated source
       format instruction;

    b) a set of data transformation routines arranged to transform data from a source
       format instruction into the appropriate parts of each target format instruction
       provided by the corresponding translation template; and

    c) a set of analysis routines arranged to identify data dependencies in a template for
       generating data for use by a code scheduler in scheduling the execution of
       translated code on said target processor.

17. A binary code translator according to claim 16 arranged to operate dynamically at the run
time of an application program being emulated.